

**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of generating a new path to a destination node in a virtual environment comprising a plurality of nodes, the method comprising:

storing nodal information identifying one or more nodes associated with a previously created path to said destination node;

dynamically reconfiguring the topology of the virtual environment to define a start node for said new path; and

processing said stored nodal information to determine the new path to said destination by including at least one node of said previously created path,

wherein when said previous path is no longer actively navigated along, an exit point is determined for said previous path, and

wherein said virtual environment is dynamically changed to include a node on said previous path, if said exit point does not comprise a node on said previous path.

2. (Cancelled)

3. (Currently Amended) A method as claimed in ~~claim 2~~ claim 1, wherein said exit point comprises a node on said previous path.

4. (Cancelled)

5. (Previously Presented) A method as claimed in claim 1, wherein in said step of processing, a return path to said previously created path which intersects said previously created path at the nearest node to said start node for said new path is determined, and said new path includes said return path.

6. (Original) A method as claimed in claim 5, wherein said nearest node is determined by processing in parallel return paths to all nodes on said previous path.

7. (Previously Presented) A method as claimed in claim 3, wherein said new path includes all subsequent nodes of said previous path between said exit point and said destination.

8. (Previously Presented) A method as claimed in claim 3, wherein in said step of processing, a return path to said previously created path which intersects said previously created path at the exit point is determined, and said new path includes said return path.

9. (Previously Presented) A method as claimed in claim 1, wherein in said step of processing said stored nodal information to determine the new path to said destination by including at least one node of said previously created path, said nodal information is processed to increase the interest component of the nodes of said previous path.

10. (Previously Presented) A method as claimed in claim 1, wherein the navigation mode employed to navigate along said previously generated path is the same as the navigation mode employed to navigate along said new path.

11. (Original) A method as claimed in claim 10, wherein a user is guided along said previous path in an automatic navigation mode, and wherein said user is able to switch to a manual navigation mode to create a manually navigated path to said start node of said new path,

and wherein said user is guided along said new path to said destination in an automatic navigation mode.

12. (Original) A method as claimed in claim 11, wherein said participant is able to switch between automatic navigation modes and manual navigation modes of their own volition at any point in said virtual environment.

13. (Original) A method as claimed in claim 10, wherein, when said participant switches from said automatic navigation mode to said manual navigation mode, an exit node is created along said previously created path.

14. (Currently Amended) A method of generating a return path to a portion of a previously created path in a virtual environment, wherein a user is able to be automatically navigated along the previously created path from a start node to a destination node, the virtual environment supporting the ability of the user to switch at their own volition from an automatic navigation mode along said previously created path to a manual navigation mode generating a second path which deviates from said previously created path in said virtual environment, the method comprising:

storing information identifying said start node, destination node and any intermediate nodes of said previously created path in said virtual environment;

dynamically reconfiguring the topology of the virtual environment to define a second start node at the point where said second path terminates; and

processing said nodal stored information to determine a return path from said second start node which intersects said previously created path at a re-entry node,

wherein when said previous path is no longer actively navigated along, an exit point is determined for said previous path, and

wherein said virtual environment is dynamically changed to include a node on said previous path, if said exit point does not comprise a node on said previous path.

15. (Original) A method as claimed in claim 14, wherein an exit node is defined at a point along said first path where said second path commences, and said step of storing information comprises storing information identifying said exit node, start node, destination node and any intermediate nodes of said first path in said virtual environment.

16. (Currently Amended) A method as claimed in claim 15, further comprising ~~performing the step of~~ validating said first path using said stored information, and if said first path is still valid, ~~performing the step of~~ also processing said nodal stored information to determine a return path from said second start node which intersects said first path at a re-entry node.

17. (Previously Presented) A method as claimed in claim 14, further comprising automatically navigating the user along the return path and the continued portion of said first path to said destination.

18. (Previously Presented) A method as claimed in claim 14, wherein in said step of processing said stored information, a parallel node search is performed from said second start node for all saved nodes of said first path to determine a return path from said second start node to a saved node of said first path.

19. (Previously Presented) A method as claimed in claim 14, wherein the closest saved node of the first path is determined as said re-entry node.

20. (Previously Presented) A method as claimed in claim 16, wherein the exit node of the first path is determined as said re-entry node.

21. (Previously Presented) A method as claimed in claim 1, wherein said user is a participant in said virtual environment.

22. (Currently Amended) A system arranged to support the generation of a new path to a destination node in a virtual environment comprising a plurality of nodes, the system comprising:

means to store nodal information identifying one or more nodes associated with a previously created path to said destination node;

means to dynamically reconfigure the topology of the virtual environment to define a start node for said new path; and

processing means to process said stored nodal information to determine the new path to said destination by including at least one node of said previously created path,

wherein when said previous path is no longer actively navigated along, an exit point is determined for said previous path, and

wherein said virtual environment is dynamically changed to include a node on said previous path, if said exit point does not comprise a node on said previous path.

23. (Currently Amended) A system arranged to support the generation of a return path to a portion of a previously created path in a virtual environment, wherein a user is able to be automatically navigated along the previously created path from a start node to a destination node, the virtual environment supporting the ability of the user to switch at their own volition from an automatic navigation mode along said previously created path to a manual navigation mode generating a second path which deviates from said previously created path in said virtual environment, the system comprising:

means to store information identifying said start node, destination node and any intermediate nodes of said previously created path in said virtual environment;

means to dynamically reconfigure the topology of the virtual environment to define a second start node at the point where said second path terminates; and

processing means to process said nodal stored information to determine a new path including return path from said second start node which intersects said previously created path at a re-entry node,

wherein when said previous path is no longer actively navigated along, an exit point is determined for said previous path, and

wherein said virtual environment is dynamically changed to include a node on said previous path, if said exit point does not comprise a node on said previous path.

24. (Previously Presented) A system as claimed in claim 22, wherein the system further comprises means to enable a user to be automatically navigated along said previously created path.

25. (Original) A system as claimed in claim 24, wherein the system further comprises means to enable a user to switch from an automatic navigation model along said previously created path to a manually navigation mode and back to an automatic navigation along said new path.

26. (Original) A user interface for a computer system as claimed in claim 25, wherein the user interface is arranged to enable a user to switch from an automatic navigation model along said previously created path to a manually navigation mode and back to an automatic navigation along said new path.

27. (Currently Amended) A computer-readable storage medium containing a computer program comprising instructions for causing one or more processors to perform the method according to claim 1 when the instructions are executed by the processor or processors.

28. (Cancelled)

29. (Currently Amended) A computer readable storage medium carrying computer readable code representing instructions for causing one or more processors to operate as the system according to claim 22 when the instructions are executed by the processor or processors.

30. - 31. (Cancelled)

32. (Currently Amended) A computer-readable storage medium carrying computer readable code representing instructions for causing one or more processors to ~~perform the method~~ operate as the system according to claim 1 when the instructions are executed by the processor or processors.

33. (New) A computer-readable storage medium containing computer readable code representing instructions for causing one or more processors to perform the method of claim 14 when the instructions are executed by the processor or processors.